

13.(Amended) A computer-implemented method[of augmenting a control flow graph in support of hierarchical path profiling, the control flow graph having an outer region and an inner region, the method] comprising:

providing a control flow graph of a program, the graph having an inner region and an outer region;

[identifying]selecting a representative entry node for the inner region;

replacing the inner region with the representative entry node;

for each prolog node of the inner region, adding an edge from the prolog node to the representative entry node; and

for each epilog node of the inner region, adding an edge from the representative entry node to the epilog node.

18.(Amended) A computer-implemented method[of augmenting a control flow graph in support of profiling a hierarchical path within a software function, the method] comprising:

[identifying]selecting a representative path within an inner region of[the] a software function, the representative path being identified by a representative entry node and a representative exit node;

for each prolog node of the inner region, adding an edge from the prolog node to the representative entry node; and

for each epilog node of the inner region, adding an edge from the representative exit node to the epilog node.

24.(Amended) A machine readable medium including instructions for a method[of augmenting a control flow graph in support of profiling a hierarchical path within a software function, the method] comprising:

[identifying]selecting a representative path within an inner region of[the] a software function, the representative path being identified by a representative entry node and a representative exit node;

for each prolog node of the inner region, adding an edge from the prolog node to the representative entry node; and

for each epilog node of the inner region, adding an edge from the representative exit node to the epilog node.

26.(Amended) A computer-implemented method[for instrumenting software in support of hierarchical path profiling] comprising:

at an entry to an inner region of a hierarchical software path, inserting an instruction to save an outer path sum;

within the inner region, inserting instructions to sum edge values into an inner path sum;
and

[at]after an exit from the inner region, inserting an instruction to increment a counter addressed as a function of the inner path sum.

27.(Amended) The computer-implemented method of claim 26 further comprising:

[at]after the exit from the inner region, inserting an instruction to restore the outer path sum.

29.(Amended) A machine readable medium including instructions for a method[of instrumenting software in support of hierarchical path profiling, the method] comprising:

at an entry to an inner region of a hierarchical software path, inserting an instruction to save an outer path sum;

within the inner region, inserting instructions to sum edge values into an inner path sum;
and

[at]after an exit from the inner region, inserting an instruction to increment a counter addressed as a function of the inner path sum.

30.(Amended) The machine readable medium of claim 29, the method further comprising:
[at]after the exit from the inner region, inserting an instruction to restore the outer path sum.

Please add the following new claims:

31. A computer-implemented method comprising:
defining an outer region of code for a computer program;
defining an inner region of the code embedded within the outer region;
assigning inner edge values within the inner region such that the inner edge values have different sums for different paths through the nodes of the inner region;
assigning outer edge values within the outer region such that the outer edge values have different sums for different paths through the nodes of the outer region, where the inner region is treated as a single node;
while executing the outer region, accumulating an outer path sum of the outer edge values;
in connection with entering the inner region,
saving the outer path sum,
accumulating an inner path sum of the inner edge values;
in connection with exiting the inner region,
modifying a profile indicator that represents the frequency of execution of a path within the inner region,
restoring the outer path sum.

32. The computer-implemented method of claim 31 wherein the code is executable code.

33. The computer-implemented method of claim 31 wherein the inner path sum is uniquely determined for different execution paths in the inner region.

34. The computer-implemented method of claim 33 further comprising addressing the profile indicator with the accumulated inner path sum.
35. The computer-implemented method of claim 31 further comprising initializing the inner path sum before accumulating it.
36. The computer-implemented method of claim 31 further comprising repeating the operations of claim 31 wherein the outer region is an inner region embedded in a further outer region.
37. The computer-implemented method of claim 36 wherein saving an outer path sum comprises pushing the outer path sum onto a stack.
38. The computer-implemented method of claim 36 wherein restoring the outer path sum comprises popping the outer path sum from a stack.
39. A machine-readable medium including instructions to carry out the method of claim 31.

REMARKS

Applicant has reviewed the Office Action mailed on October 4, 2002, and the references cited therewith.

This Amendment amends previously pending claims 1, 13, 18, 24, 26, 27, 29, and 30, and adds new claims 31-39. As a result, claims 1-39 are now pending in the Application.

Claim Rejections

Claims 13, 15-16, 18-21, and 24-25 were rejected under 35 U.S.C. §102(b) as being anticipated by Muchnick, "Advanced Compiler Design Implementation", pages 169-177. Applicant respectfully traverses these rejections.

The cited paper to Muchnick describes ways for drawing a control-flow graph of a computer program that follows the code of that program. Applicant, however, modifies the